Robust trajectory planning of autonomous vehicles at intersections with communication impairments

Neha Chohan

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 12.08.2019

Supervisor and advisor

Prof. Themistoklis Charalambous

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical
Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Neha Chohan

Title Robust trajectory planning of autonomous vehicles at intersections with communication impairments

Degree programme Space Science and Technology

Major Space Robotics and Automation **Code of major** ELEC3047

Supervisor and advisor Prof. Themistoklis Charalambous

Date 12.08.2019 **Number of pages** 33 **Language** English

Abstract

In this thesis, we consider the trajectory planning of an autonomous vehicle to cross an intersection within a given time interval. The vehicle communicates its sensor data to a central coordinator which then computes the trajectory for the given time horizon and sends it back to the vehicle. We consider a realistic scenario in which the communication links are unreliable, the evolution of the state has noise (e.g., due to the model simplification and environmental disturbances), and the observation is noisy (e.g., due to noisy sensing and/or delayed information). The intersection crossing is modeled as a chance constraint problem and the stochastic noise evolution is restricted by a terminal constraint. The communication impairments are modeled as packet drop probabilities and Kalman estimation techniques are used for predicting the states in the presence of state and observation noises. A robust sub-optimal solution is obtained using convex optimization methods which ensures that the intersection is crossed by the vehicle in the given time interval with very low chance of failure.

Keywords Intersection crossing; robust trajectory planning; unreliable communications, stochastic model predictive control

Preface

I want to thank my supervisor and mentor, Prof. Themistoklis Charalambous, for his excellent guidance and advice throughout the course of this thesis. He made this research enriching and interesting and pushed me to put in my best efforts. Secondly, I thank my entire research group (Distributed and Networked Control Systems), for their constant support, review and suggestions at all stages of my work. I also want to thank all the administrative and support staff at the School of Electrical Engineering for ensuring a smooth functioning of all related processes. Lastly, I thank my family and friends for being the wind beneath my wings, and for making all of this possible.

Otaniemi, 12.08.2019

Neha Chohan

Contents

Notations

Symbols

$\mathbb{N}$	set of natural numbers
$\mathbb{R}$	set of real numbers
$\mathbb{N}_0^T$	$\Big\{0, 1, \ldots, T\Big\}$
$A \succ 0$	positive definite matrix (for a square matrix $A \in \mathbb{R}^{n \times n}$)
$\mathbb{1}$	all-ones vector (of appropriate dimensions)
$\mathbb{0}$	all-zeros vector (of appropriate dimensions)
I	identity matrix (of appropriate dimensions)

Operators

$\text{diag}\{A\}$	matrix with diagonal entries only
$\mathbb{E}\{\cdot\}$	expectation of the argument
$\text{tr}\{\cdot\}$	trace of the argument
$\text{cov}(\cdot)$	covaraince of the argument
$\|\| \cdot \|\|$	the 2-norm of the argument
$\text{blkdiag}(A_1, \ldots, A_N)$	the block diagonal matrix

1 Introduction

The expected advent of autonomous vehicles in the near future opens up a number of possibilities for efficient driving through communication and coordination. This, among others, can help mitigate the issues of road safety and traffic congestion as well as boost fuel efficiency.

Road safety should be a high priority issue, since about 1.24 million people die and 50 million are hurt in road accidents each year [1]. In addition, drivers and passengers waste around 90 billion hours in traffic jams per year. Traffic conditions in major cities have become an important issue, since congestion results in time delays, carbon dioxide emissions, higher energy expenditure and higher accident risks (see, for example, [2] and references therein). Furthermore, in some major cities, as much as a third of the petrol used is burned by people looking for a space to park, contributing to additional traffic congestion with all its consequences. The ultimate goal of the European Union is to reduce greenhouse gases (in 2015 the target was to reduce it by about 20% until 2020, as per the Paris Agreement), which means that the transport efficiency has to be drastically improved.

Apart from improving the efficiency of vehicles, a promising approach to all these issues is put forth by the foreseeable potential of automated vehicles to replace human driving in the near future. Many new and established companies are pushing to bring truly driverless cars to market as early as 2020. The first large-scale deployments will be in fleets, where companies own the vehicles and provide revolutionary new services like robot taxis and driverless delivery. The technology is still quite expensive, but prices will fall quickly as economies of scale takeover. However, a lot is still needed to be done to bring about the transformational changes we expect driverless cars to bring to our society. The most important of these is to ensure that there are proper protocols and rules to gain optimal efficiency from driverless cars while ensuring near-perfect safety from them. The current work is a step towards this direction.

A significant proportion of accidents take place near road intersections, which are among the most complex regulated traffic subsystems. The modeling of autonomous vehicles crossing at intersections is a crucial problem in this area. With efficient and effective algorithms in place, it is possible to completely do away with traffic signals and fixed predefined schedules for vehicle crossings. Indeed, by developing intelligent algorithms where crossing vehicles exchange information and mutually decide on a safe schedule, one can improve the traffic flow, increase safety and save fuel [3–6].

This problem is usually broken down into three parts: 1) determining an optimal priority list for the concerned vehicles around the intersection, 2) finding optimal time intervals for each of these vehicles to cross the intersections, and 3) determining an optimal trajectory for each vehicle to cross the intersection in its appointed time interval.

In the current work, we consider the third sub-part of the intersection crossing problem. Additionally, we introduce real life complexities of communication losses and noise in state and observation. Communication losses can be modeled in several ways. Two distinct considerations are: channels with and without temporal correlation (memory). In channels without memory, the current state is an independent and

identically distributed (i.i.d.) process. In channels with memory, the current state of the channels depends on, and is influenced by the previous state(s). In modeling channels with memory, various methods such as Markov chain model, Gilbert-Elliot model and their variations are available; see, e.g., [7–9]. In this thesis, we show the behavior of both channels with and without memory for the communication losses in the system. The noise evolution is usually of a probabilistic nature, in particular, a Gaussian distribution. For dealing with probabilistic noise, stochastic model predictive control (SMPC) strategies are usually implemented. Their usage and importance has been elaborated well in [10]. The history and development of SMPC is documented extensively in [11, 12]. A good use of SMPC technique for navigation of autonomous vehicle is detailed in [13]. In SMPC techniques, the constraints are modeled either as chance constraints or be satisfied by the expectation (see [14–17]) . For this thesis, we use chance constraints for the intersection crossing objective.

The rest of this report is organized as follows. In Section 2, a detailed description and analysis of the work done so far in this area is provided. In Section 3, the main objective of this work is emphasized and all the methods and models, necessary for the development of this work, are explained. The main results are presented in Section 4. The performance of our proposed approach is evaluated in Section 5. Finally, in Section 6, we draw conclusions and discuss possible future directions. Results of this thesis work have been published at: (1) 57th Annual Allerton Conference on Communication, Control, and Computing, hosted by the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign, USA (Accepted), (2) IEEE Transactions on Intelligent Transportation Systems (Submitted).

2 Background

Various methods have been suggested for dealing with the first part of the problem including approximation methods [18], heuristics-based methods [19] and other algorithms [20, 21]. Similarly, a good amount of research has been carried out where the first part is assumed to be known and the second part of the problem is addressed [22–25]. In [26] and [27], a combination of the above problem are addressed simultaneously. In most of these papers, two assumptions are common: perfect communication and a deterministic system model. Both aforementioned assumptions are far from reality and, hence, it is necessary to develop a framework where these assumptions are lifted in order to model a more realistic scenario.

Coming to the current problem under consideration, *i.e.*, determining a trajectory for a single vehicle to cross the intersection within a predetermined time interval. One of the earlier works in this regard is done in [28], where a chance constraint is used for determining an optimal trajectory for a stochastic system. However, here the problem is solved analytically using LQG model and hence no hard constraints or communication impairments are considered. A more recent work [29], considers the vehicle to be driven by human drivers instead of AVs and aims at computing speed advice for each vehicle using convex optimization. It considers the driver's response speed and lag time as a Gaussian distribution and optimizes the best control strategy over an ensemble. This can be considered as one of the first steps before going towards complete autonomy. However, here also, the communication between the vehicles is assumed to be perfect and all states are deterministic.

Okamoto *et al.* in [30] provide a theoretical framework for optimal control of the covariance of stochastic finite discrete-time linear systems subject to chance constraints. This is cast as an MPC problem and solved using convex optimization techniques with the optimization variable being the control gain matrix. The usually separable problems of mean and covariance problem are, in this case, linked together by chance constraints. However, the observation is assumed to be noiseless, and no communication impairments are considered. Nazari *et al.* in [31] take a step in the direction of a stochastic system with imperfect communication links for a single vehicle crossing the intersection. It uses chance-constrained MPC model and gives the control input matrix using Gaussian approximation. However, the process covariance is allowed to evolve unrestricted as per the system dynamics and the observation covariance is zero (perfectly observable state). Moreover, the packet drop probability is modeled as an independent and identically distributed (i.i.d.) process, ignoring the temporal correlation of the channel variation. In [32], instead of assuming known moments of the i.i.d., for the process and observation noise, they are estimated through samples of the stochastic quantities, which is a step towards more realistic considerations.

However, as rightly stated in [33], errors encountered in real communication channels often appear in clusters rather than randomly. Hence, the previous state of the channel greatly influences the next state. In order to take advantage of this dependence to predict the packet drops more accurately, communication channels are modeled to have memory. In [7], the channel is modeled as a Markov chain for

establishing the dependency between adjacent states. A more complex modeling is done using the Gilbert-Elliot model of communication channels as elaborated in [34] and [35]. A comparison between Markov and Gilbert-Elliot modeling is given in [9], which shows that the 3-state Markov chain and Gilbert-Elliot model reproduce the packet losses and burst length of a real channel equally well. In this work, however, we use the 2-state Markov chain to model channel with memory since it is sufficient for the assumed packet losses in this system.

3 Method and Modeling

In this work, we consider a stochastic, finite-horizon discrete-time linear vehicle system with both process and observation noises, modeled as Gaussian distributions. Moreover, we model communication impairments between the vehicle and central coordinator using packet loss probability distribution for channels both with and without temporal correlation (memory). Here, the packet drops from vehicle sensor to central coordinator are taken into consideration. In case of packet drops from controller to vehicle actuator, the previously communicated control input sequence from the model predictive controller would be implemented. Hence, these packet drops are not separately addressed. Kalman estimation method is used to predict the evolution of the state taking into account the noise and packet drops. The objective of crossing the intersection is modeled as a chance constraint. In addition, we have a limit on the terminal covariance value in order to ensure efficient control input requirements. While we follow the basic framework from [30] for model dynamics and terminal covariance constraints, these are modified to account for communication impairments and observation noises. In particular, we extend the current model to account for estimating the state and also change the covariance propagation model to allow for communication losses.

3.1 Dynamic Model

The stochastic, finite-horizon, discrete, linear dynamics of the vehicle is modeled as follows:

$$x_{k+1} = A_k x_k + B_k u_k + D_k w_k, \tag{1a}$$

$$y_{k+1} = C_{k+1} x_{k+1} + G_{k+1} r_{k+1}, \quad k \in \mathbb{N}_0^{N-1}, \tag{1b}$$

where $x_k \in \mathbb{R}^{n_x}$, in this case $n_x = 3$, since x_k is a three dimensional vector representing the position, velocity and acceleration of the vehicle, *i.e.*, $x_k = \begin{bmatrix} s_k & v_k & a_k \end{bmatrix}^T$, $u_k \in \mathbb{R}$ is the control input, $y_k \in \mathbb{R}^{n_x}$ is the output observation of the system, and $w_k, r_k \in \mathbb{R}^{n_x}$ are zero mean white Gaussian process and observation noise, respectively, with unit covariance. The time step k ranges from 0 to $N-1$, where N is the total number of time steps considered. Matrices A_k and B_k are defined as in [36] as

$$A_k = A = \begin{bmatrix} 1 & h & 0 \\ 0 & 1 & h \\ 0 & 0 & 1 - \frac{h}{\tau} \end{bmatrix}, B_k = B = \begin{bmatrix} 0 \\ 0 \\ \frac{h}{\tau} \end{bmatrix},$$

where h is the discrete-time interval and τ is a vehicle-specific parameter called the acceleration-to-deceleration ratio. In this model, we assume all output states are measurable *i.e.* $C_k = I$ for all k. Process noise matrix D_k and observation noise matrix G_k can be modeled according to the given scenario properties. Note that, currently, we assume the position state to be just a one-dimensional value along the length of the road. The lateral position is assumed to be constant.

Our cost function to be minimized is given by:

$$J(x_0, \ldots, x_{N-1}, u_0, ..., u_{N-1}) = \mathbb{E}\left[\sum_{k=0}^{N-1} \left(\tilde{x}_k^T Q_k \tilde{x}_k + u_k^T R_k u_k\right)\right],$$

where $\tilde{x}$ is the deviation from the optimal trajectory, $k \in \mathbb{N}_0^{N-1}$, $Q_k \succeq 0$ and $R_k \succ 0$. There is no terminal cost since the end state is constrained as explained in the subsequent sections.

The initial state x_0 is a random vector of Gaussian distribution given by known mean and covariance matrices (μ_0, Σ_0) *i.e.*, $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$. We assume that the end state can be approximated by a Gaussian distribution of given mean and covariance matrices (μ_N, Σ_N), *i.e.*, $x_N \sim \mathcal{N}(\mu_N, \Sigma_N)$.

3.2 Communication Model: Memoryless Channel

First, we consider channels which have no memory of their previous state, but are only aware of their current state. Under such communication, the output observation of the system, y_{k+1}, is communicated to the central coordinator subject to a packet drop probability p_d. If $p_d = 0$, there are no packet drops and all observations are communicated from the vehicle to the central coordinator. If $p_d = 1$, then all packets get dropped and there is no observation being communicated by the vehicle. We consider two ways of modeling this packet drop probability, as follows:

3.2.1 Sample of Probability Distribution

We introduce the success indicator term, δ_k, which is 1 if the observation data has been communicated to the vehicle, and 0 otherwise. The value of δ_k is obtained according to the packet drop probability p_d. We consider several different packet drop probabilities between and including 0 and 1. A sample of $\delta = \left[\delta_1, \delta_2, ...\delta_N\right]$ is generated based on the packet drop probability p_d.

3.2.2 Direct Probability Distribution

In this case, instead of binary δ_k values, we consider the probability distribution value itself to determine the amount of observation communicated back to the vehicle. Hence the distribution p_d itself is taken in the corresponding equations.

3.3 Communication Model: Channel with Memory

Here, we model the packet losses in channel with memory as a 2-state Markov chain model. We consider two states of the channel: good(G) and bad (B); see Fig. 1. In the good state (G), the packet is communicated successfully, whereas in the bad state (B), the packet is lost. The probability of transitioning from the good state to bad state is given by p and thus the probability of remaining in good state is $1 - p$. Similarly, the probability of transitioning from bad state to good state is given by q, and thus the probability of remaining in the bad state is $1 - q$.

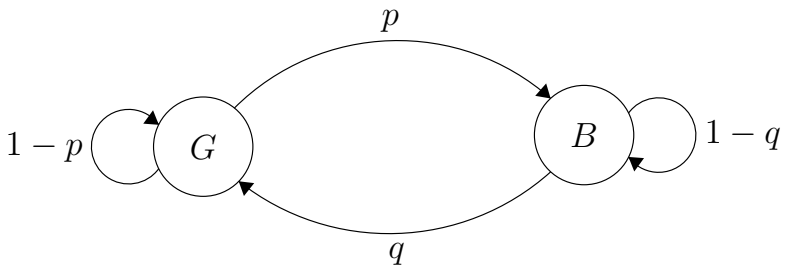


Figure 1: Two-state Markov chain model

In this paper, we assume that the probabilities p and q are known to us. This is possible to obtain from the channel packet loss statistics over a long period of usage. Knowing the above probabilities, we again derive an ensemble δ for packet drops over the entire horizon, as it was the case in Section 3.2.1, but in this case of the channel with memory.

3.4 Kalman State Estimation

The Kalman estimator is used to get the best estimation of the state taking into account process noise, observation noise and packet drops. Let the Kalman gain be denoted by F_k. The covariance prediction and update using Kalman estimation based on [37] is calculated as follows:
The *a priori* covariance estimation is

$$\Sigma_{k+1|k} = A\Sigma_{k|k}A + D_k D_k^T. \tag{2}$$

The Kalman gain update:

$$F_{k+1} = \Sigma_{k+1|k}(G_{k+1}G_{k+1}^T + \Sigma_{k+1|k})^{-1}.$$

The *a posteriori* covariance update is given by

$$\Sigma_{k+1|k+1} = (I - \gamma_{k+1}F_{k+1})\Sigma_{k+1|k}, \tag{3}$$

where $\gamma_k = \delta_k$ in case of sampled distribution and $\gamma_k = p_d$ in case of direct probability distribution. The *a priori* and *a posteriori* estimates of the state based are given as

$$\begin{aligned}\hat{x}_{k+1|k} &= A\hat{x}_{k|k} + Bu_k,\ k \in \mathbb{N}_0^{N-1},\\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + \gamma_{k+1}F_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}),\end{aligned}$$

respectively.

Let $e_k \triangleq x_k - \hat{x}_{k|k}$, thus e_k represents the state estimation error as a Gaussian white noise. Then, the state equation can be rewritten as:

$$\begin{aligned}x_{k+1} &= \hat{x}_{k+1|k+1} + e_{k+1}\\ &= \hat{x}_{k+1|k} + \gamma_{k+1}F_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}) + e_{k+1}\\ &= \hat{x}_{k+1|k} + \gamma_{k+1}F_{k+1}(Cx_{k+1} + G_{k+1}r_{k+1} - C\hat{x}_{k+1|k}) + e_{k+1}.\end{aligned}$$

Hence,

$$(I - \gamma_{k+1}F_{k+1}C)(x_{k+1} - \hat{x}_{k+1|k}) = \gamma_{k+1}F_{k+1}G_{k+1}r_{k+1} + e_{k+1}.$$

Rearranging and substituting for the *a priori* estimate,

$$x_{k+1} = A(x_k - e_k) + Bu_k + (I - \gamma_{k+1}F_{k+1}C)^{-1}(\gamma_{k+1}F_{k+1}G_{k+1}r_{k+1} + e_{k+1}). \quad (4)$$

Thus, writing (4) in the form of (1a), we get

$$x_{k+1} = Ax_k + Bu_k + D'_k w'_k,$$

where

$$D'_k = \begin{bmatrix} -A & (I - \gamma_{k+1}CF_{k+1})^{-1} & (I - \gamma_{k+1}CF_{k+1})^{-1}(\gamma_{k+1}F_{k+1}G_{k+1}) \end{bmatrix}$$

and

$$w'_k = \begin{bmatrix} e_k \\ e_{k+1} \\ r_{k+1} \end{bmatrix}, \quad k \in \mathbb{N}_0^{N-1}.$$

We can, therefore, represent both the state estimation error and observation noise within a single noise quantity. This representation will be useful in the subsequent analysis.

3.5 Preliminary Equations

For a given intersection to be crossed, say at position s_{exit}, a vehicle modeled with (1a)-(1b) needs a number of time steps, say N. The compact system model containing the equations at every time instant, from time step 0 until time step N, stacked together can be written as:

$$\chi = \mathcal{A}x_0 + \mathcal{B}\upsilon + \mathcal{D}\omega, \quad (5)$$

where

$$\chi = \begin{bmatrix} x_0^T & x_1^T & \dots & x_N^T \end{bmatrix}^T \in \mathbb{R}^{(N+1)n_x},$$
$$\upsilon = \begin{bmatrix} u_0^T & u_1^T & \dots & u_{N-1}^T \end{bmatrix}^T \in \mathbb{R}^{Nn_u},$$
$$\omega = \begin{bmatrix} w_0'^T & w_1'^T & \dots & w_{N-1}'^T \end{bmatrix}^T \in \mathbb{R}^{Nn_w},$$

and the matrices $\mathcal{A} \in \mathbb{R}^{(N+1)n_x \times n_x}$, $\mathcal{B} \in \mathbb{R}^{(N+1)n_x \times Nn_u}$, $\mathcal{D} \in \mathbb{R}^{(N+1)n_x \times Nn_w}$ are defined as $\mathcal{A} = \begin{bmatrix} I & \bar{A}_1 & \bar{A}_2 & \cdots & \bar{A}_N \end{bmatrix}^T$, where $\bar{A}_k = A_{k-1,0} = A_{k-1}A_{k-2}\dots A_0$,

$$\mathcal{B} = \begin{bmatrix} 0 & \bar{B}_1 & \cdots & \bar{B}_{N-1} \end{bmatrix}^T$$
$$= \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ B_0 & 0 & 0 & \dots & 0 \\ B_{1,0} & B_1 & 0 & \dots & 0 \\ B_{2,0} & B_{2,1} & B_{2,1} & \dots & 0 \\ \vdots & & & \ddots & \\ B_{N-1,0} & B_{N-2,1} & B_{N-3,1} & \dots & B_{N-1} \end{bmatrix},$$

where $B_{k1,k0} = A_{k1,k0+1}B_{k0}$, thus,

$$\mathcal{B} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ B_0 & 0 & 0 & \dots & 0 \\ AB_0 & B_1 & 0 & \dots & 0 \\ A^2B_0 & AB_1 & B_2 & \dots & 0 \\ \vdots & & & \ddots & \\ A^{N-1}B_0 & A^{N-2}B_1 & A^{N-3}B_2 & \dots & B_{N-1} \end{bmatrix}.$$

and,

$$\mathcal{D} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ D'_0 & 0 & 0 & \dots & 0 \\ AD'_0 & D'_1 & 0 & ... & 0 \\ A^2D'_0 & AD'_1 & D'_2 & \dots & 0 \\ \vdots & & & \ddots & \\ A^{N-1}D'_0 & A^{N-2}D'_1 & A^{N-3}D'_2 & \dots & D'_{N-1} \end{bmatrix}.$$

Hence, the objective function can be rewritten as

$$J(\chi, \upsilon) = \mathbb{E}\left[(\chi - \bar{\chi})^T \bar{Q}(\chi - \bar{\chi}) + \upsilon^T \bar{R}\upsilon\right], \tag{6}$$

where $\bar{\chi}$ is the desired trajectory of the vehicle, $\bar{Q} = \text{blkdiag}(Q_1, Q_2, \dots, Q_{N-1}, 0)$ and $\bar{R} = \text{blkdiag}(R_1, R_2, \dots, R_{N-1})$.

We assume, for now, that the observation gives the true state, *i.e.*, the observation noise, $r_k = 0$. Thus, the state feedback controller can be defined in the form

$$\upsilon_k = l_k \left[\mathbb{1}^T, x_0^T, , x_1^T, ..., x_k^T\right]^T,$$

where $l_k \in \mathbb{R}^{n_u \times n_x(k+2)}$. Thus, the relationship between χ and υ can be written as

$$\boldsymbol{\upsilon} = L\boldsymbol{\chi},$$

where $\boldsymbol{\chi} = [\mathbb{1}^T, \chi^T]^T \in \mathbb{R}^{n_x(N+2)}$ is the augmented state sequence until step N, $\boldsymbol{\upsilon} \in \mathbb{R}^{Nn_u}$ is the control matrix until step $N-1$ and $L \in \mathbb{R}^{Nn_u \times (N+2)n_x}$ is the control gain matrix. L matrix is required to be causal and hence takes the form $L = \left[L_1, L_{\boldsymbol{\chi}}\right]$, where $L_1 \in \mathbb{R}^{Nn_u \times n_x}$ and $L_{\boldsymbol{\chi}} \in \mathbb{R}^{Nn_u \times (N+1)n_x}$ is a lower triangular block matrix. From (5), we can write the new state equation as:

$$\boldsymbol{\chi} = \begin{bmatrix} I & \mathbb{0} \\ \mathbb{0} & \mathcal{A} \end{bmatrix} \boldsymbol{\chi}_0 + \begin{bmatrix} \mathbb{0} \\ \mathcal{B} \end{bmatrix} L\boldsymbol{\chi} + \begin{bmatrix} \mathbb{0} \\ \mathcal{D} \end{bmatrix} \boldsymbol{\omega}, \tag{7}$$

where $\boldsymbol{\chi}_0 = [\mathbb{1}^T, x_0^T]^T$ and $\boldsymbol{\omega} = [\mathbb{0}^T, \omega^T]^T$. Rewriting (7) we get,

$$\boldsymbol{\chi} = (I - \mathbf{B}L)^{-1}(\mathbf{A}\boldsymbol{\chi}_0 + \mathbf{D}\boldsymbol{\omega}), \tag{8}$$

where $\mathbf{A} = \text{blkdiag}(I, \mathcal{A})$, $\mathbf{B} = [\mathbb{0}, \mathcal{B}]^T$ and $\mathbf{D} = [\mathbb{0}, \mathcal{D}]^T$.

We define a new variable $K \triangleq L(I - \mathbf{B}L)^{-1}$, where $K = \begin{bmatrix} K_1, K_{\boldsymbol{\chi}} \end{bmatrix}$, with $K_1 \in \mathbb{R}^{Nn_u \times n_x}$ and $K_{\boldsymbol{\chi}} \in \mathbb{R}^{Nn_u \times (N+1)n_x}$, is a lower triangular block matrix. Substituting K into (8), we get the following equations for state and control matrices:

$$\boldsymbol{\chi} = (I + \mathbf{B}K)(\mathbf{A}\boldsymbol{\chi}_0 + \mathbf{D}\boldsymbol{\omega}), \tag{9}$$

$$\boldsymbol{\upsilon} = K(\mathbf{A}\boldsymbol{\chi}_0 + \mathbf{D}\boldsymbol{\omega}). \tag{10}$$

The variable to be optimized and tuned in this problem is K or equivalently, L.

4 Main Results

4.1 Cost function

Redefining the initial state mean and covariance as

$$\boldsymbol{\mu}_0 = \begin{bmatrix} \mathbb{1} \\ \mu_0 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_0 = \begin{bmatrix} \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \Sigma_0 \end{bmatrix}, \tag{11}$$

respectively, we can now define the desired trajectory (i.e., the trajectory that is followed when there are no uncertainties and the initial state is given by the initial state mean), denoted by $\bar{\boldsymbol{\chi}}$ as,

$$\bar{\boldsymbol{\chi}} = (I + \mathbf{B}K)\mathbf{A}\boldsymbol{\mu}_0. \tag{12}$$

Substituting (9), (10) and (12) in objective function (6) and knowing that $\mathbb{E}[\boldsymbol{\chi}_0] = \boldsymbol{\mu}_0$, $\mathbb{E}[\boldsymbol{\chi}_0\boldsymbol{\chi}_0^T] = \boldsymbol{\mu}_0\boldsymbol{\mu}_0^T + \boldsymbol{\Sigma}_0$ and $\mathbb{E}[\boldsymbol{\chi}_0\boldsymbol{\omega}^T] = 0$, the objective function can be re-written as

$$J(K) = \mathrm{tr}(((I+\mathbf{B}K)^T\bar{Q}(I+\mathbf{B}K)+K^T\bar{R}K)(\mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^T+\mathbf{D}\mathbf{D}^T)+K^T\bar{R}K\mathbf{A}\boldsymbol{\mu}_0\boldsymbol{\mu}_0^T\mathbf{A}^T).$$

This is a quadratic expression in K.

4.2 Covariance Matrix

The covariance of the closed loop state matrix $\boldsymbol{\chi}$ is by definition given as

$$\boldsymbol{\Sigma}_{\boldsymbol{\chi}} = \mathbb{E}\left[(\boldsymbol{\chi} - \bar{\boldsymbol{\chi}})(\boldsymbol{\chi} - \bar{\boldsymbol{\chi}})^T\right].$$

Substituting (9) and (12), we get

$$\boldsymbol{\Sigma}_{\boldsymbol{\chi}} \quad = \quad \mathbb{E}\left[((I + \mathbf{B}K)(\mathbf{A}(\boldsymbol{\chi}_0 - \boldsymbol{\mu}_0) + \mathbf{D}\boldsymbol{\omega}))((I + \mathbf{B}K)(\mathbf{A}(\boldsymbol{\chi}_0 - \boldsymbol{\mu}_0) + \mathbf{D}\boldsymbol{\omega}))^T\right].$$

Simplifying and rewriting, we get,

$$\boldsymbol{\Sigma}_{\boldsymbol{\chi}} = (I + \mathbf{B}K)(\mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^T + \mathbf{D}\mathbf{D}^T)(I + \mathbf{B}K)^T. \tag{13}$$

Let $\boldsymbol{\Sigma}_{op} \triangleq \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^T + \mathbf{D}\mathbf{D}^T$, be the matrix denoting open loop covariance dynamics. Hence,

$$\boldsymbol{\Sigma}_{op} = \begin{bmatrix} I \\ \bar{A}_1 \\ \bar{A}_2 \\ \vdots \\ \bar{A}_N \end{bmatrix} \boldsymbol{\Sigma}_0 \begin{bmatrix} I & \bar{A}_1{}' & \bar{A}_2{}' & \dots & \bar{A}_N{}' \end{bmatrix} + \mathbf{D}\mathbf{D}^T$$

$$= \begin{bmatrix} I\Sigma_0I' & I\Sigma_0A_1' & I\Sigma_0A_2' & I\Sigma_0A_3' & \dots & I\Sigma_0A_N' \\ A_1\Sigma_0I' & A_1\Sigma_0A_1' & A_1\Sigma_0A_2' & A_1\Sigma_0A_3' & \dots & A_1\Sigma_0A_N' \\ A_2\Sigma_0I' & A_2\Sigma_0A_1' & A_2\Sigma_0A_2' & A_2\Sigma_0A_3' & \dots & A_2\Sigma_0A_N' \\ A_3\Sigma_0I' & A_3\Sigma_0A_1' & A_3\Sigma_0A_2' & A_3\Sigma_0A_3' & \dots & A_3\Sigma_0A_N' \\ \vdots & & & & \ddots & \\ A_N\Sigma_0I' & A_N\Sigma_0A_1' & A_N\Sigma_0A_2' & A_N\Sigma_0A_3' & \dots & A_N\Sigma_0A_N' \end{bmatrix} + \mathbf{D}\mathbf{D}^T.$$

For no communication losses, this can be written in terms of covariance evolution at each step given as defined in equation (2) given by a new variable,

$$\mathbf{\Sigma}_{op}^{nl} = \begin{bmatrix} \Sigma_0 & \Sigma_0 A' & \Sigma_0 A_2' & \Sigma_0 A_3' & \dots & \Sigma_0 A_N' \\ A\Sigma_0 & \Sigma_1 & \Sigma_1 A_1' & \Sigma_1 A_2' & \dots & \Sigma_0 A_{N-1}' \\ A_2\Sigma_0 & A_1\Sigma_1 & \Sigma_2 & \Sigma_2 A_1' & \dots & \Sigma_2 A_{N-2}' \\ A_3\Sigma_0 & A_2\Sigma_1 & A_1\Sigma_2 & \Sigma_3 & \dots & \Sigma_3 A_{N-3}' \\ \vdots & & & & \ddots & \\ A_N\Sigma_0 & A_{N-1}\Sigma_1 & A_{N-2}\Sigma_2 & A_{N-3}\Sigma_3 & \dots & \Sigma_N \end{bmatrix}.$$

Now consider the case with communication packet losses. The value of each of the Σ_k in the above equation changes to that given by equation (3). Thus, we can define a new variable,

$$\mathbf{\Sigma}_{op}^{l} = \begin{bmatrix} \Sigma_0 & \Sigma_0 A' & \Sigma_0 A_2' & \Sigma_0 A_3' & \dots & \Sigma_0 A_N' \\ A\Sigma_0 & \Sigma_1 & \Sigma_1 A_1' & \Sigma_1 A_2' & \dots & \Sigma_0 A_{N-1}' \\ A_2\Sigma_0 & A_1\Sigma_1 & \Sigma_2 & \Sigma_2 A_1' & \dots & \Sigma_2 A_{N-2}' \\ A_3\Sigma_0 & A_2\Sigma_1 & A_1\Sigma_2 & \Sigma_3 & \dots & \Sigma_3 A_{N-3}' \\ \vdots & & & & \ddots & \\ A_N\Sigma_0 & A_{N-1}\Sigma_1 & A_{N-2}\Sigma_2 & A_{N-3}\Sigma_3 & \dots & \Sigma_N \end{bmatrix}, \tag{14}$$

where the Σ_k values are given by (3). Thus, we can write the covariance of state matrix $\boldsymbol{\chi}$ in the presence of packet losses as:

$$\mathbf{\Sigma}_{\boldsymbol{\chi}} \triangleq (I + \mathbf{B}K)\mathbf{\Sigma}_{op}^{l}(I + \mathbf{B}K)^T. \tag{15}$$

4.3 Chance constraints for intersection crossing

Due to the stochastic nature of the system, putting hard constraints on the intersection crossing within a given time might result in infeasible solutions. Hence we model this as a chance constraint with very low probability of failure. The required chance constraint is given by

$$Pr(s_N \leq s_{\text{exit}}) \leq p_f,$$

where p_f is a predefined low probability of failure. Rewriting this constraint in terms of $\boldsymbol{\chi}$ and using appropriate multiplicative matrices α and β, we get,

$$Pr(\alpha^T\bar{\boldsymbol{\chi}} \leq \beta) = \Phi\left(\frac{\beta - \alpha^T\bar{\boldsymbol{\chi}}}{\sqrt{\alpha^T\mathbf{\Sigma}_{\boldsymbol{\chi}}\alpha}}\right) \leq p_f, \tag{16}$$

where Φ is the cumulative distribution function of the standard Gaussian distribution. Now, let

$$\Sigma_p = \alpha^T\mathbf{\Sigma}_{\boldsymbol{\chi}}\alpha = \alpha^T(I + \mathbf{B}K)\mathbf{\Sigma}_{op}^{l}(I + \mathbf{B}K)^T\alpha.$$

From (14), we know that $\mathbf{\Sigma}_{op}^{l}$ is symmetric, although not necessarily positive definite. We define M such that $M \triangleq (\mathbf{\Sigma}_{op}^{l})^{\frac{1}{2}}$. If $\mathbf{\Sigma}_{op}^{l}$ is positive definite, M is real, otherwise M is complex. However in both cases, M is symmetric. Thus, we can rewrite,

$$\Sigma_p = \alpha^T\mathbf{\Sigma}_{\boldsymbol{\chi}}\alpha = \alpha^T(I + \mathbf{B}K)M^TM(I + \mathbf{B}K)^T\alpha.$$

Now, $M(I + \mathbf{B}K)^T\alpha$ is a vector. Let us denote it by $\bar{u}$. Thus $\Sigma_p = \bar{u}^T\bar{u}$. Rewriting (16), we get,

$$\beta - \alpha^T\bar{\boldsymbol{\chi}} - \sqrt{\Sigma_p}\Phi^{-1}(p_f) \leq 0.$$

Substituting for Σ_p,

$$\beta - \alpha^T\bar{\boldsymbol{\chi}} - \sqrt{\bar{u}^T\bar{u}}\Phi^{-1}(p_f) \leq 0.$$

Now, $\bar{u}^T\bar{u} = \lambda_{\max}(\bar{u}^T\bar{u})$ since $\bar{u}^T\bar{u}$ is a $1x1$ matrix. Hence,

$$\beta - \alpha^T\bar{\boldsymbol{\chi}} - \sqrt{\lambda_{\max}(\bar{u}^T\bar{u})}\Phi^{-1}(p_f) \leq 0.$$

Now, $\lambda_{\max}(\bar{u}^T\bar{u}) \leq \lambda_{\max}(\bar{u}^H\bar{u})$ where the equality holds for real matrices. Since $\Phi^{-1}(p_f) < 0$ for $p_f < 0.5$, the following inequality holds:

$$\beta - \alpha^T\bar{\boldsymbol{\chi}} - \sqrt{\lambda_{\max}\bar{u}^T\bar{u}}\Phi^{-1}(p_f) \leq \beta - \alpha^T\bar{\boldsymbol{\chi}} - \sqrt{\lambda_{\max}\bar{u}^H\bar{u}}\Phi^{-1}(p_f) \leq 0. \tag{17}$$

By definition of 2-norm for matrices,

$$\beta - \alpha^T\bar{\boldsymbol{\chi}} - ||\bar{u}||\Phi^{-1}(p_f) \leq 0.$$

Resubstituting for $\bar{u}$, the final form of the chance constraint is,

$$\beta - \alpha^T(I + \mathbf{B}K)\mathbf{A}\boldsymbol{\mu}_0 - ||M(I + \mathbf{B}K)^T\alpha||\Phi^{-1}\big(p_f\big) \leq 0. \tag{18}$$

Remark 1 *Inequality (17) holds under the assumption that the failure probability, $p_f < 0.5$, i.e., the term $\Phi^{-1}(p_f)$ is negative. At $p_f = 0.5$, $\Phi^{-1}(p_f) = 0$, which eliminates the nonlinear part (i.e., the covariance term) from the equation and only the mean value is left. For $p_f > 0.5$, $\Phi^{-1}(p_f)$ becomes positive and this inequality no longer holds. However, in real life scenarios, the system is designed such that the failure probability is much smaller than 0.5, and, hence, our assumption is reasonable.* ⌟

4.4 Constraint on terminal covariance

We constrain the evolution of the process covariance to be less than a predefined value Σ_{lim}. It provides a narrow distribution of the end state and thus the predicted states are closer to the mean μ_N. This ensures that the μ_N required for the success of the chance constraint is not too far away from the required s_{exit}. The constraint is modeled as follows:

$$E_N\boldsymbol{\Sigma}_{\boldsymbol{\chi}}E_N^T \leq \Sigma_{\text{lim}},$$

where $E_N \triangleq [0, 0, 0, \ldots, I] \in \mathbb{R}^{n_x \times (N+2)n_x}$ is the appropriate multiplying matrix to get the N^{th} time step covariance.
Hence the constraint becomes,

$$E_N(I + \mathbf{B}K)\boldsymbol{\Sigma}_{op}^l(I + \mathbf{B}K)^TE_N^T \leq \Sigma_{\text{lim}}.$$

Since by assumption, $\Sigma_{\text{lim}} \geq 0$, the above inequality can be rewritten as:

$$I - (\Sigma_{\text{lim}}^{-1/2})^TE_N(I + \mathbf{B}K)\boldsymbol{\Sigma}_{op}^l(I + \mathbf{B}K)^TE_N^T\Sigma_{\text{lim}}^{-1/2} \geq 0.$$

Substituting for $\mathbf{\Sigma}^l_{op}$,

$$I - (\Sigma_{\text{lim}}^{-1/2})^T E_N (I + \mathbf{B}K) M^T M (I + \mathbf{B}K)^T E_N^T \Sigma_{\text{lim}}^{-1/2} \geq 0.$$

Being symmetric, the matrix

$$\Xi \triangleq (\Sigma_{\text{lim}}^{-1/2})^T E_N (I + \mathbf{B}K) M^T M (I + \mathbf{B}K)^T E_N^T \Sigma_{\text{lim}}^{-1/2}$$

is diagonalizable via the orthogonal matrix $S \in \mathbb{R}^{n_x \times n_x}$. Thus, $S(I - \text{diag}(\lambda_1, \ldots, \lambda_{n_x})S^T \geq 0$ where $\lambda_1, \ldots, \lambda_{n_x}$ are the eigenvalues of Ξ. The last inequality is implied by $1 - \lambda_{\max}(\Xi) \geq 0$. Again, since $\lambda_{\max}(M^T M) \leq \lambda_{\max}(M^H M)$, we get,

$$\lambda_{\max}(\Xi) \leq \lambda_{\max}((\Sigma_{\text{lim}}^{-1/2})^T E_N (I + \mathbf{B}K) M^H M (I + \mathbf{B}K)^T E_N^T \Sigma_{\text{lim}}^{-1/2}).$$

Hence a stronger constraint on the covariance is:

$$1 - \lambda_{\max}((\Sigma_{\text{lim}}^{-1/2})^T E_N (I + \mathbf{B}K) M^H M (I + \mathbf{B}K)^T E_N^T \Sigma_{\text{lim}}^{-1/2}) \geq 0. \tag{19}$$

By definition of 2-norm, we can rewrite constraint (19) as:

$$1 - \left\| M (I + \mathbf{B}K)^T \mathbf{E}_N^T \Sigma_{\text{lim}}^{-1/2} \right\|^2 \geq 0. \tag{20}$$

This is the final form of the constraint on the terminal covariance of the state.

4.5 Input constraints

In any real vehicle scenario, the power input from the engine and consequently, the available acceleration to the vehicle is limited. This needs to be accounted while designing the solution. To this end, in this paper, we restrict the available input acceleration to the vehicle based on generic car acceleration limits as follows:

$$u_{\min} \leq u \leq u_{\max}. \tag{21}$$

4.6 Optimization problem

Based on the system model and all the constraint considerations, the final optimization problem can be defined as follows:

$$\begin{aligned} &\min_K \ J(K) \\ &\text{s.t. } (18), (20), (21). \end{aligned} \tag{22}$$

The result of this optimization gives a robust control input for each step in the given horizon to cross the intersection successfully in the presence of all the aforementioned constraints. The control input of the first step can be implemented by the vehicle and the entire process can be repeated at the next time step, as per the MPC methodology. This control input is communicated by the central coordinator back to the vehicle. In case of communication loss or infeasibility at a particular step, the previously computed input scheme can be implemented till a new version is available. For the proof of convexity of the above optimization problem, see [30].

5 Numerical Examples

In this section, we demonstrate the performance of the proposed algorithm through numerical examples and, thereby, quantify its advantages over previous methods. Problem (22) is coded in MATLAB and run using the CVX toolset for disciplined convex optimization [38].

A horizon of $N = 20$ time steps with time period, $h = 0.2s$ is considered. A vehicle starts from $s_0 = 0\,m$ with velocity $v = 5m/s$ and acceleration $a = 0\,m/s^2$. This is taken as the initial mean of the state $u_0 = \begin{bmatrix} 0 & 5 & 0 \end{bmatrix}$. It is required that it crosses the intersection given by $s_{\text{exit}} = 30m$ in the given time horizon. Since $s_{\text{exit}} = 30m$, the vehicle needs to accelerate from its current speed to a higher value to cross the intersection in the given horizon. The initial covariance of the state is assumed to be $\Sigma_0 = \text{diag}\begin{Bmatrix} 1 & 0.1 & 0.1 \end{Bmatrix}$. The noise matrices for state, is taken as $D = \text{diag}\begin{Bmatrix} 0.5 & 0.05 & 0.05 \end{Bmatrix}$ and for observer noise, $G = 0.5D$. Vehicle specific parameters are taken as $\tau = 10$ and input acceleration limits are $\begin{bmatrix} -5, & 3 \end{bmatrix} m/s^2$. The cost matrices for state and input costs are considered are $Q = \text{diag}\begin{Bmatrix} 1 & 0.01 & 5 \end{Bmatrix}$ and $R = 5$, respectively.

5.1 Memoryless Channel Results

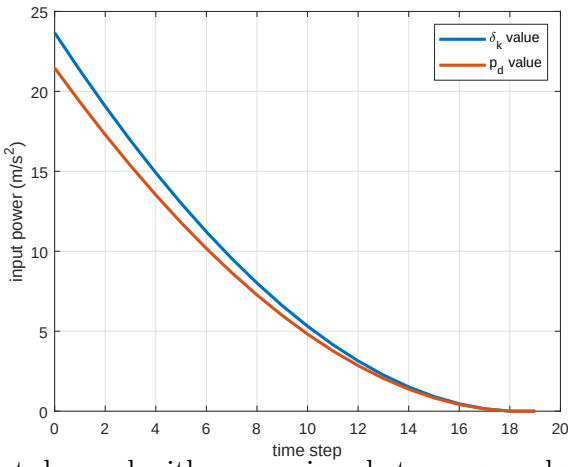


Figure 2: Control input demand with comparison between sampled (δ_k) distribution and direct probability (p_d) distribution with $p_d = 0.5$ and $p_f = 0.0005$.

In this scenario, we have considered two methods of modeling the impairments: *(i)* one by taking an ensemble of the packet drop probability distribution, and *(ii)* another by taking the probability distribution values themselves. It is observed that the method with the ensemble of the packet drop probability distribution warrants a more stringent demand for control and is also closer to the real scenario (a comparison is given in Fig. 2.) As a result, for all the subsequent simulation we will be using an ensemble of the packet drop probability distribution.

5.1.1 Open-loop control

In what follows, the trajectory is predicted for $N = 20$ time steps by the convex optimization problem (22) and the corresponding control sequence is generated (open-loop control).

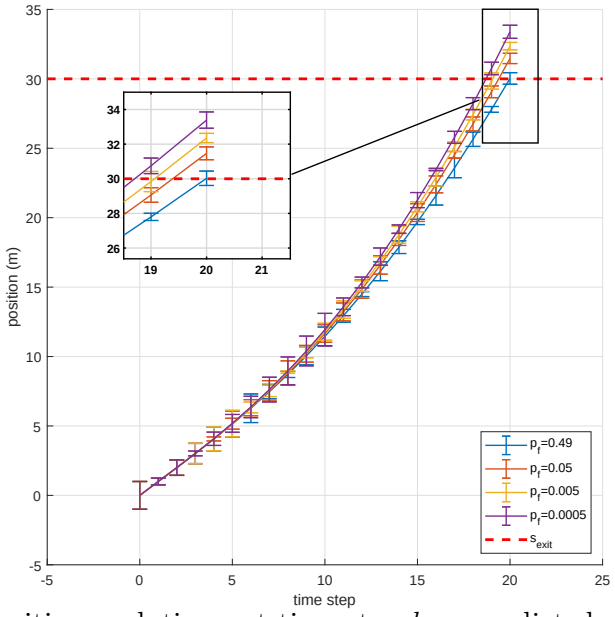
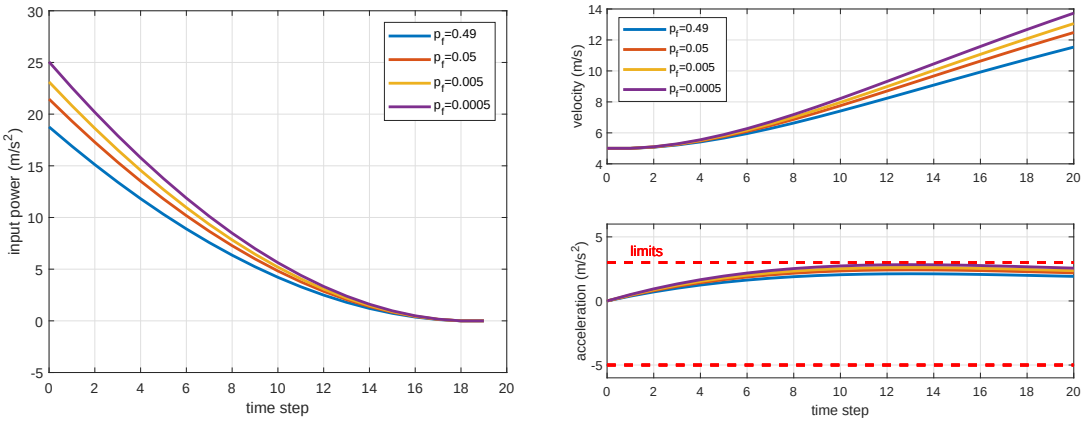


Figure 3: Vehicle position evolution wrt time step k as predicted at $k = 0$ until $k = N$ with varying p_f and constant $p_d = 0.5$. The length of the errorbar above and below the mean, as depicted in the figure, is the standard deviation $1\sigma_k = \sqrt{\Sigma_k}$.



(a) Control input

(b) Velocity and acceleration

Figure 4: Evolution wrt time step k as predicted at $k = 0$ until $k = N$ with varying p_f and constant $p_d = 0.5$.

First, the packet drop probability is kept constant at $p_d = 0.5$ and the probability of failure is varied from $p_f = \big[0.49, 0.05, 0.005, 0.0005\big]$. From the plot of vehicle position (see Fig. 3), we can see that, at $p_f = 0.49$ (blue curve), the position mean is

very close to $s = 30\,m$ and the covariance (errorbar) is distributed such that there is approximately 50% chance that the vehicle crosses the intersection.

As we keep decreasing the probability of failure, the mean and the corresponding covariance bar move further away from the s_{exit} position, thereby ensuring that the vehicle crosses the intersection with more and more certainty. For all practical purposes, we will assume $p_f = 0.0005$, thereby ensuring that the vehicle crosses the intersection with a very high level of certainty. The control input requirement (see Fig. 4a) increases as we decrease the probability of failure and this is expected, since control becomes increasingly conservative. However, since we cannot risk the failure of the vehicle to cross the intersection, we allow the higher input requirement. The velocity and acceleration of the vehicle (see Fig. 4b) also increases as failure probability decreases. However, for the current noise matrix and initial covariance, the acceleration is well within the constraint bounds.

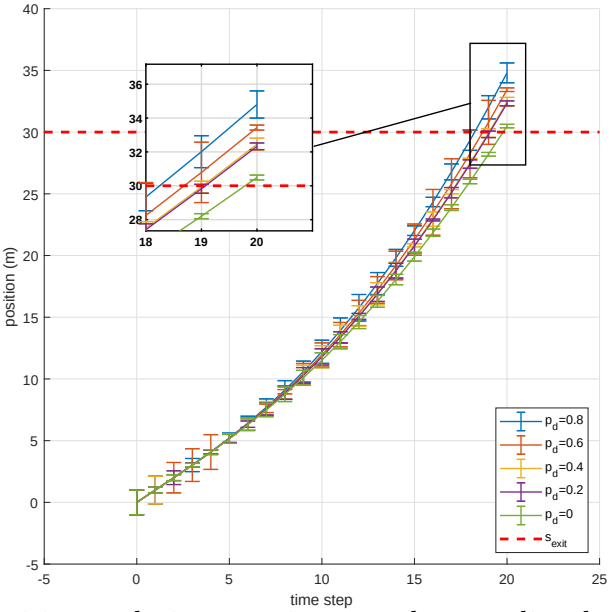
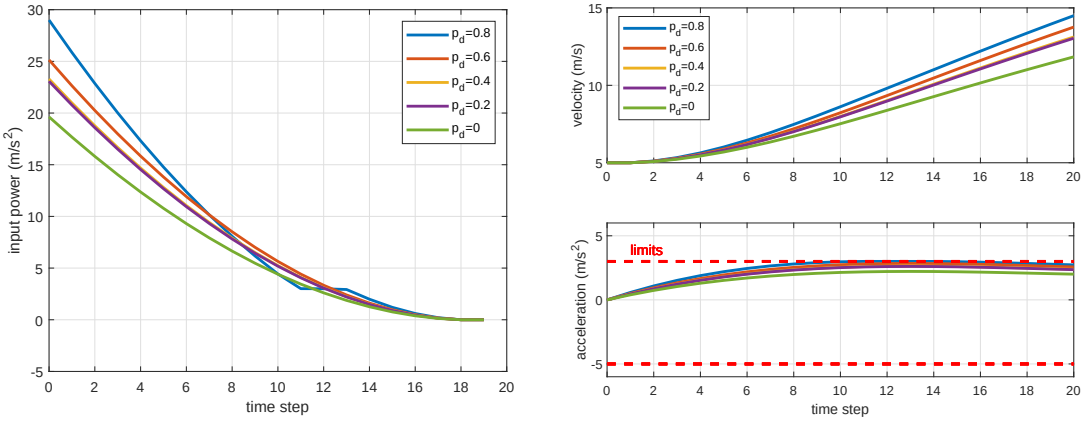


Figure 5: Vehicle position evolution wrt time step k as predicted at $k = 0$ until $k = N$ with varying p_d and constant $p_f = 0.0005$.

In the next set of results, the failure probability is kept constant at $p_f = 0.0005$ and packet drop probabilities considered are $p_d = \begin{bmatrix} 0.0, & 0.2, & 0.4, & 0.6, & 0.8 \end{bmatrix}$. Beyond $p_d = 0.8$, the optimization becomes infeasible because of the high packet drops since the covariance cannot be kept within the required limits. Consider Fig. 5 which gives the position of the vehicle. As the packet drop probability increases, the noise in the system increases and hence the covariance gets bigger.

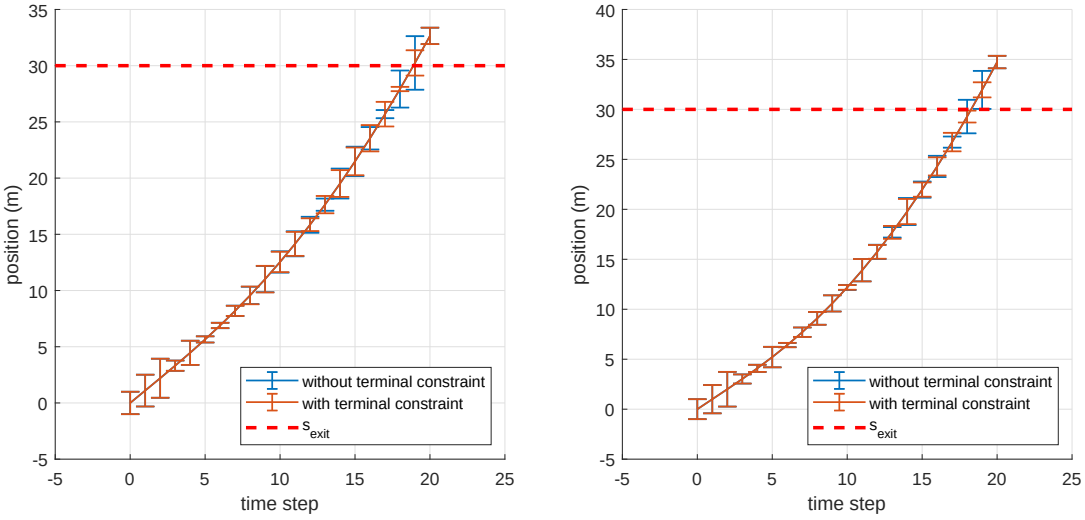
The difference between the covariance when $p_d = 0$ (green bars) and $p_d = 0.8$ (blue bars) is starkly visible. As the covariance gets bigger, the mean position also moves further upward from s_{exit} so as to ensure that the probability of failure stays within the bounds. As it is shown in Fig. 6a, the control input also increases as the packet drop probability increases. This is because, packet losses mean that the covariance evolves as an open loop without any feedback from the sensors. Hence the input



(a) Control input (b) Velocity and acceleration

Figure 6: Evolution wrt time step k as predicted at $k = 0$ until $k = N$ with varying p_d and constant $p_f = 0.0005$.

has to be made increasingly conservative to account for this higher covariance. The velocity and acceleration of the vehicle also increase with the packet drop probability, as illustrated in Fig. 6b. Here, at $p_d = 0.8$, the acceleration touches the upper bound; however, the vehicle is under control.



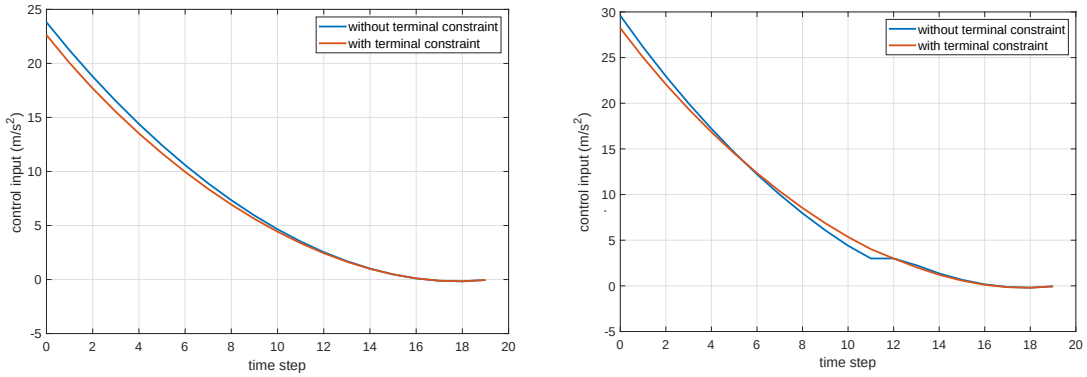
(a) $p_d = 0.8$ and $p_f = 0.05$. (b) $p_d = 0.8$ and $p_f = 0.0005$.

Figure 7: Vehicle position evolution with and without the terminal covariance constraints with different p_f

Next, the advantage of having the terminal constraint on the covariance evolution is illustrated. This is done by considering two cases and comparing the results with and without the terminal constraint. In both cases, the packet drop probability is taken as $p_d = 0.8$ because that's where the effect of the terminal constraint is most apparent. In first case, a high probability of failure $p_f = 0.05$ is taken to emphasize the difference in the mean value of the final position. Fig. 7a shows that the presence of the terminal constraint forces the end constraint to be within a certain limit

and hence the mean position is closer to s_{exit} compared to no constraint case. This constraint also affects all the previous evolution of the covariance and hence the overall control demand is reduced by this (see Fig. 8a).

In the second case, the practical failure probability $p_f = 0.0005$ is taken. In this case, the position difference is not as high (Fig. 7b), but the effect on control input is clearly seen (Fig. 8b). We observe that in the case with no terminal constraint, the control input is high in the beginning, reaches the limit and gets saturated. On the other hand, with terminal constraint in effect, the control input gets adjusted from the beginning itself to ensure smooth input and thus maintain acceleration within the limits. However, it is noted that, at higher noise levels, when the terminal covariance constraint cannot be satisfied, an infeasibility is generated, whereas the case with no constraint still gives a feasible solution. A way to work around this problem would be to check for infeasibility with the constrained solution and if it is infeasible, re-solve the problem without the terminal constraint.



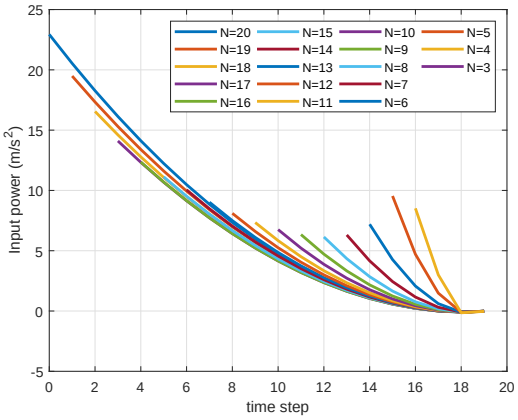
(a) $p_d = 0.8$ and $p_f = 0.05$. (b) $p_d = 0.8$ and $p_f = 0.0005$.

Figure 8: Control input evolution with and without the terminal covariance constraints with different p_f

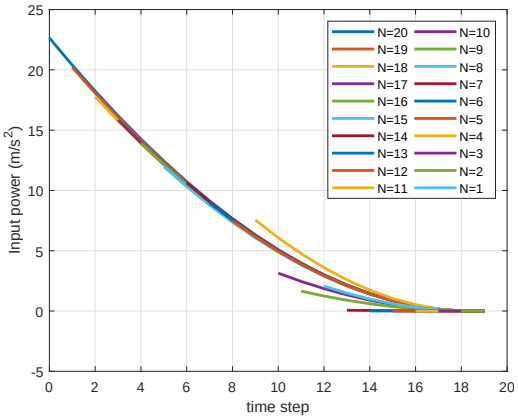
Remark 2 *It is observed that the terminal covariance constraint does not get activated for most of the simulated numerical values. It is reached only in the case of some extreme initial state values and very high packet drop probabilities. Similarly, the input constraints also get activated only in extreme cases. Hence it is reasonable to approximate the end state with a Gaussian distribution.* ⌟

In formulating the convex constraints as in (17), we see that the transpose product of the vectors is replaced by the hermitian product whose norm always has an equal or higher value. It means that we are using a more stringent condition in the constraint. Hence, the corresponding solution is sub-optimal. We can quantify this deviation from the optimal value by comparing the value used in the constraint ($||M(I + \mathbf{B}K)^T\alpha||$) and the actual standard deviation of the noise at the N^{th} time step. For example, in the standard case with $p_f = 0.0005$ and $p_d = 0.5$, this constraint value is 0.8531 whereas the standard deviation of the noise is 0.3754. Hence, the solution is approximately 2.3 times more conservative, in this case.

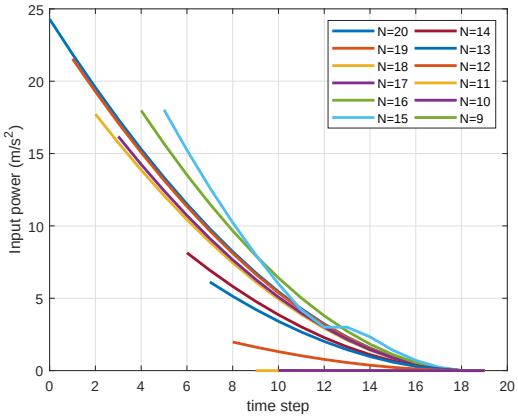
5.1.2 Close-loop control



(a) Direct probability distribution.



(b) Same ensemble for each loop.



(c) Different ensemble for each loop.

Figure 9: Control sequence generated at each time step of the receding horizon with $p_d = 0.5$ and $p_f = 0.0005$

A aforementioned, in Section 5.1.1, results have been run with $N = 20$, *i.e.*, the trajectory is predicted for 20 time steps using optimization problem (22) and the corresponding control sequence is generated (open-loop control). Following the MPC method, however, the control at one time step is to be executed and the entire problem should be re-evaluated for the next step. This is how the vehicle would run the algorithm in practice.

We simulate this closed loop modeling in MATLAB by taking the first predicted state in the previous loop as the initial condition for the next loop in which the horizon is for $N - 1$ steps. Simulations using both the direct probability distribution and an ensemble of the distribution were done. In the direct distribution system (Fig. 9a), we see that the control changes significantly at each step. This is because, for the same probability distribution, the algorithm calculates a very different strategy depending on the number of steps available. In using the ensemble, we can either use the same ensemble for every loop so that the control strategy remains the same (Fig. 9b) or take a new sample for each loop and have a varying control strategy

(Fig. 9c). The latter may result in higher costs since the previously planned control strategy may need to be changed completely for the new ensemble. The above mentioned plots (Figs. 9a, 9b, 9c) show that our algorithm can be used recursively in the entire MPC horizon and hence is executable in the practical setup. Note that sometimes, at lower values of N, the optimization becomes infeasible and the previously calculated control input can be applied at that time step.

5.2 Channel with Memory Results

5.2.1 Open-loop control

Here, we take the probability of transition from good state to bad state, p, as 0.3 and from bad state to good state, q, as 0.6. This corresponds to a steady state

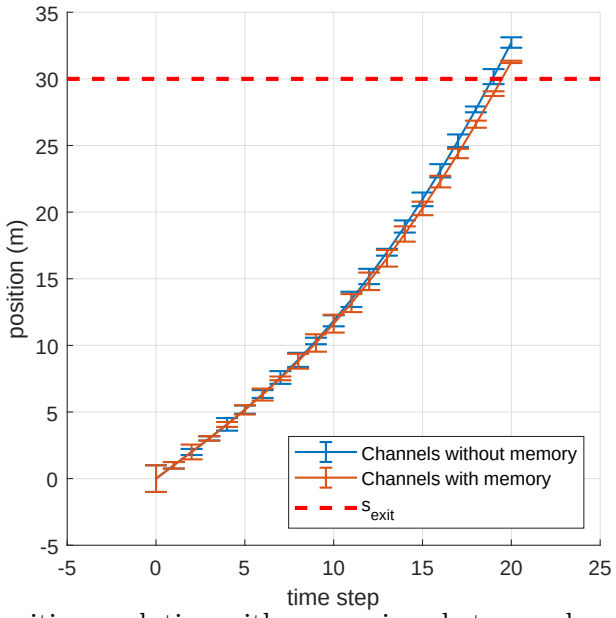
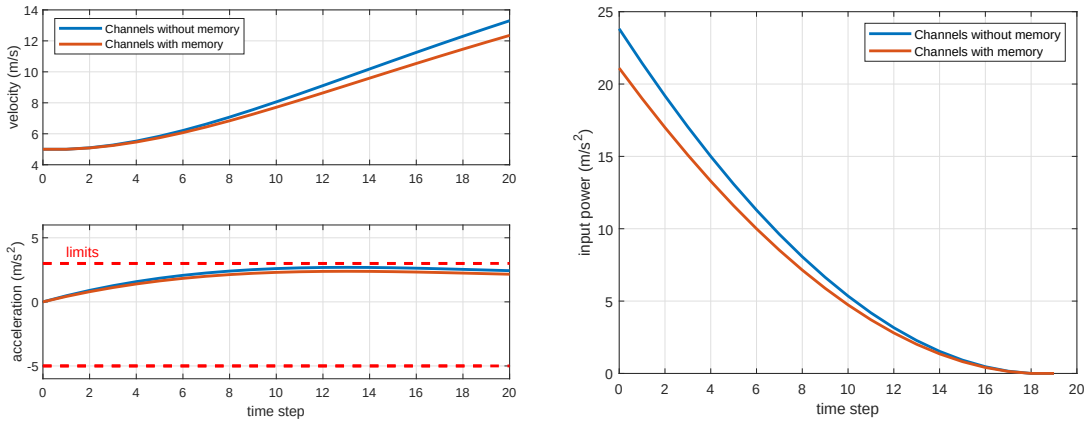


Figure 10: Vehicle position evolution with comparison between channels without memory ($p_d = 0.33$) and channels with memory (Markov model, $p = 0.3, q = 0.6$).

packet drop probability of 0.33. With these probabilities, we again get an ensemble of δ. Using $p_f = 0.0005$, we compare the results (see Figs. 10, 11a, 11b) obtained through this modeling with that obtained through memoryless channel with a packet drop probability $p_d = 0.33$ for N=20 time steps, keeping all other variables at their default values. We can see that, using the memory channel based Markov model, the input requirement and subsequent position and velocity are lesser than those with memoryless channels. Hence, channels with memory not only provide a more realistic modeling of packet drops, but also decreases the control demand as a result.

Remark 3 *A number of runs with different values of p, q were simulated and it was observed that this method shows better results than memoryless channels with corresponding steady state probabilities.* ⌟

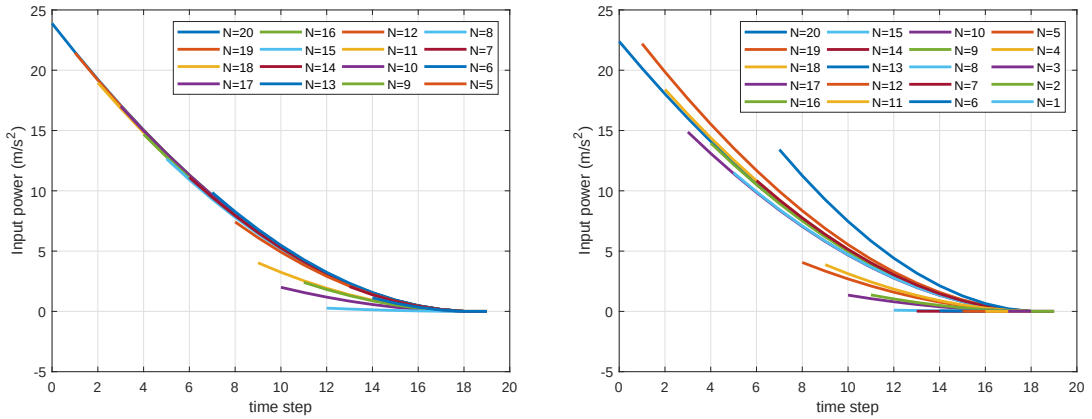


(a) Velocity and acceleration. (b) Control input

Figure 11: Evolution with comparison between channels without memory ($p_d = 0.33$) and channels with memory (Markov model, $p = 0.3, q = 0.6$).

5.2.2 Close-loop control

Like the closed loop control in memoryless channels, here also, we take the two different cases of ensemble prediction. We can either use the same ensemble for every loop so that the control strategy remains the same (Fig. 12a), or take a new sample



(a) same ensemble for each loop. (b) different ensemble for each loop.

Figure 12: Control sequence generated at each time step of the receding horizon with $p = 0.3$ and $q = 0.6$ for channels with memory.

for each loop and have a varying control strategy. The run with same ensemble gives similar results as channels without memory. However, for different ensembles case, the memoryless channels, (Fig. 9c) gave an larger spread in control strategy. However, in the case of channels with memory, (Fig. 12b), the control strategy remained more or less the same. This might be because there is relation and dependencies between the adjacent packets and hence the overall pattern tends to be similar in each run. This provides a model closer to the real scenario and hence more efficient trajectory planning.

6 Conclusions and Future Directions

6.1 Conclusions

This work has addressed the problem of determining a robust trajectory for vehicles to cross an intersection within a given time-slot in the presence of communication impairments. It was solved as a convex optimization problem with intersection crossing requirement as a chance constraint and terminal covariance constraint as a deterministic constraint. We note that the constraints used are rather conservative and, as a result, the final trajectory is a feasible one, but not (necessarily) optimal.

This work has provided a theoretical framework to include the real world scenarios of communication losses as well as noisy observations in the path planning for automated vehicles, which were in most cases so far, taken to be perfect due to difficulty in modeling the same. An algorithm is obtained, which can work with varying degrees of packet loss and noise levels to provide a feasible, sub-optimal solution with very low probability of failure in crossing the intersection within a given time frame.

6.2 Future Directions

The first future course of action would be to make the current setup even more closer to real world scenario. One way would be to introduce delays in the control action execution and find the corresponding stable control strategy. Another improvement would be to have communication channels with memory which includes the losses of individual bits in the packet (Gilbert-Elliot Models) for better estimation and real time updating of the packet drop probability.

It would also be attempted to find an optimal solution to the given problem rather than the current conservative solution. It is expected that this will require a change in the way the problem is formulated as well as a different method to solve the optimization problem.

Secondly, it is envisioned to extend the solution to find the optimal path for multiple vehicles simultaneously. A further step in this direction would be to consider the time-slots for vehicle as optimization variables and solve for the optimal crossing time for each vehicle, given a predetermined priority order. Once the above objectives have been achieved and a solid framework has been developed for them, the problem of determining the priority list for vehicles to cross the intersection can be addressed.

References

[1] "The future of cars: Wireless wheels," The Economist, Sept. 2014.

[2] A. Y. Bigazzi and M. A. Figliozzi, "Congestion and emissions mitigation: A comparison of capacity, demand, and vehicle based strategies," *Transportation Research Part D: Transport and Environment*, vol. 17, no. 7, pp. 538–547, 2012.

[3] M. R. Hafner, D. Cunningham, L. Caminiti, and D. D. Vecchio, "Cooperative Collision Avoidance at Intersections: Algorithms and Experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, Sept. 2013.

[4] H. Wymeersch, G. R. de Campos, P. Falcone, L. Svensson, and E. G. Ström, "Challenges for cooperative ITS: Improving road safety through the integration of wireless communications, control, and positioning," in *International Conference on Computing, Networking and Communications (ICNC)*, Feb. 2015, pp. 573–578.

[5] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of Cooperative Autonomous Vehicles: Toward safer and more efficient road transportation," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, Nov. 2016.

[6] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2016.

[7] K. K. Lee and S. T. Chanson, "Packet loss probability for real-time wireless communications," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, pp. 1569–1575, Nov. 2002.

[8] F. Liu, T. H. Luan, X. S. Shen, and C. Lin, "Dimensioning the packet loss burstiness over wireless channels: a novel metric, its analysis and application," *Wireless Communications and Mobile Computing*, vol. 14, no. 12, pp. 1160–1175, July 2012.

[9] B. P. Milner and A. B. James, "An analysis of packet loss models for distributed speech recognition," in *INTERSPEECH*, 2004.

[10] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, Dec. 2016.

[11] A. Charnes and W. W. Cooper, "Deterministic equivalents for optimizing and satisficing under chance constraints," *Operations Research*, vol. 11, no. 1, pp. 18–39, Feb. 1963.

[12] K. J. Åström, "Introduction to stochastic control theory," *Press Inc. NY*, 1970.

[13] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge (AVEC)*, Sept. 2014.

[14] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, Aug. 1999.

[15] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, Jan. 2007.

[16] P. Li, M. Wendt, and G. Wozny, "A probabilistically constrained model predictive controller," *Automatica*, vol. 38, no. 7, pp. 1171–1176, July 2002.

[17] A. Geletu, M. Klöppel, H. Zhang, and P. Li, "Advances and applications of chance-constrained approaches to systems optimisation under uncertainty," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1209–1232, July 2013.

[18] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *IEEE American Control Conference (ACC)*, July 2015, pp. 763–768.

[19] R. Hult, M. Zanon, S. Gras, and P. Falcone, "An MIQP-based heuristic for Optimal Coordination of Vehicles at Intersections," in *IEEE Conference on Decision and Control (CDC)*, Dec. 2018.

[20] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, March 2008.

[21] F. Yan, M. Dridi, and A. E. Moudni, "An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 1, pp. 183–200, March 2013.

[22] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Energy-optimal coordination of autonomous vehicles at intersections," in *European Control Conference (ECC)*, June 2018, pp. 602–607.

[23] Y. Bichiou and H. A. Rakha, "Real-time optimal intersection control system for automated/cooperative vehicles," *International Journal of Transportation Science and Technology*, vol. 8, no. 1, pp. 1–12, March 2019.

[24] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti, "Revisiting street intersections using slot-based systems," *PLOS ONE*, vol. 11, no. 3, p. e0149607, March 2016.

[25] C. Wuthishuwong, A. Traechtler, and T. Bruns, "Safe trajectory planning for autonomous intersection management by using vehicle to infrastructure communication," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, Feb. 2015.

[26] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2018.

[27] S. G. H. W. P. F. Robert Hult, Mario Zanon, "Optimization-based coordination of connected, automated vehicles at intersections." [Online]. Available: https://research.chalmers.se/en/publication/509433

[28] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, gaussian systems," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2152–2159.

[29] A. Katriniok, S. Kojchev, E. Lefeber, and H. Nijmeijer, "A stochastic model predictive control approach for driver aided intersection crossing with uncertain driver time delay," in *European Control Conference (ECC)*, 2019.

[30] K. Okamoto, M. Goldshtein, and P. Tsiotras, "Optimal Covariance Control for Stochastic Systems Under Chance Constraints," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 266–271, April 2018.

[31] M. A. Nazari, T. Charalambous, J. Sjöberg, and H. Wymeersch, "Remote control of automated vehicles over unreliable channels," in *IEEE Wireless Communications and Networking Conference (WCNC)*, April 2018, pp. 1–6.

[32] V. Lefkopolous and M. Kamgarpour, "Using uncertainity data in chance-constrained trajectory planning," in *European Control Conference (ECC)*, 2019.

[33] L. N. Kanal and A. R. K. Sastry, "Models for channels with memory and their applications to error control," *Proceedings of the IEEE*, vol. 66, no. 7, pp. 724–744, July 1978.

[34] G. Hasslinger and O. Hohlfeld, "The gilbert-elliott model for packet loss in real time services on the internet," in *14th GI/ITG Conference - Measurement, Modelling and Evalutation of Computer and Communication Systems*, March 2008, pp. 1–15.

[35] P. Almstrom, M. Rabi, and M. Johansson, "Networked state estimation over a gilbert-elliot type channel," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, Dec. 2009.

[36] A. Colombo, M. Bahraini, and P. Falcone, "Measurement scheduling for control invariance in networked control systems," *IEEE Conference on Decision and Control (CDC)*, pp. 3361–3366, 2018.

[37] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, Sept. 2004.

[38] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.